

ARCHITETTURA DEI SISTEMI ELETTRONICI

LEZIONE N° 17

- **Reti sequenziali sincronizzate complesse**
 - **Esempio**
 - Parte operativa
 - Parte di controllo

A.S.E.

17.1

Richiami

- **Reti combinatorie**
 - **Tecnica di sintesi (minimizzazione) strutturata**
 - **Sintesi euristica**
 - Sommatore
 - Sottrattore
 -
- **Reti sequenziali**
 - **Variabili di stato**
 - **Tecnica di sintesi (minimizzazione degli stati) strutturata**
 - **Sintesi euristica**
 - Contatori
 - Shift register
 -

A.S.E.

17.2

Esempio

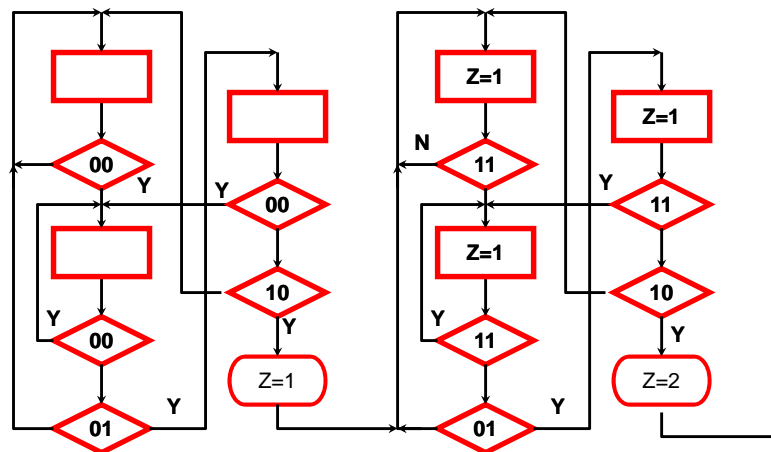
- **Riconoscitore di due sequenze in cascata**
 - Ingressi
 - X_1, X_0
 - Uscite
 - Z_3, Z_2, Z_1, Z_0
- **Descrizione**
 - prima riconosce la sequenza 00, 01, 10
 - dopo riconosce la sequenza 11, 01, 10
 - Incrementa l'uscita tutte le volte che una sequenza viene riconosciuta (modulo 16)

A.S.E.

17.3

Diagramma di flusso

{00-01-10}_ {11-01-10}



Occorrono 8 blocchi simili al seguente

A.S.E.

17.4

Osservazioni

- In totale si hanno $6 \times 8 = 48$ stati interni
- Occorrono 6 variabili di stato
 - $2^6 = 64$, potenza del 2 immediatamente superiore a 48
- Rete combinatoria a $6 + 2 = 8$ ingressi e 4 uscite
 - Non è possibile ottimizzarla manualmente
- _____
- Considerando la macchina come macchina sequenziale complessa si ha $\rightarrow\rightarrow$

A.S.E.

17.5

Macchina sequenziale complessa Parte operativa

- La parte operativa può essere un contatore modulo 16 che si incrementa quando
 - La parte di controllo ha verificato che sono passati due termini della sequenza giusta
 - e gli ingressi assumono il valore $X_1 = 1, X_0 = 0$

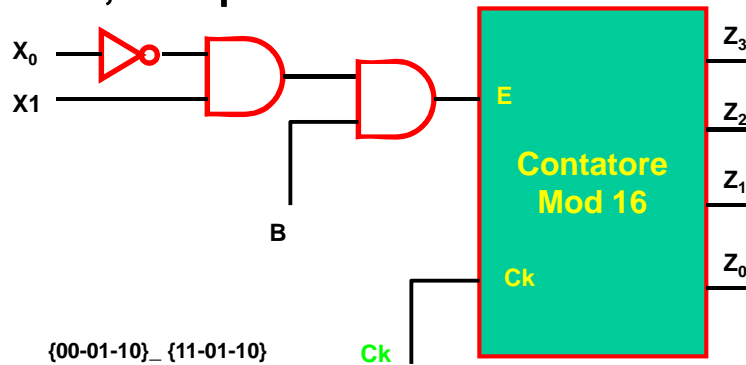
$\{00-01-10\}_\{11-01-10\}$

A.S.E.

17.6

Schema della parte operativa

- Incrementa se ($X_1 = 1$ e $X_0 = 0$) e se il segnale "B", della parte di controllo vale 1

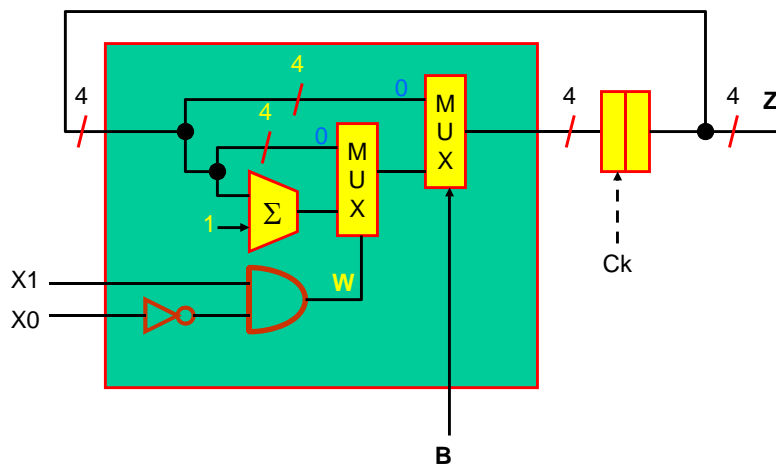


A.S.E.

17.7

Parte operativa Soluzione alternativa

Soluzione che evidenzia la rete combinatoria

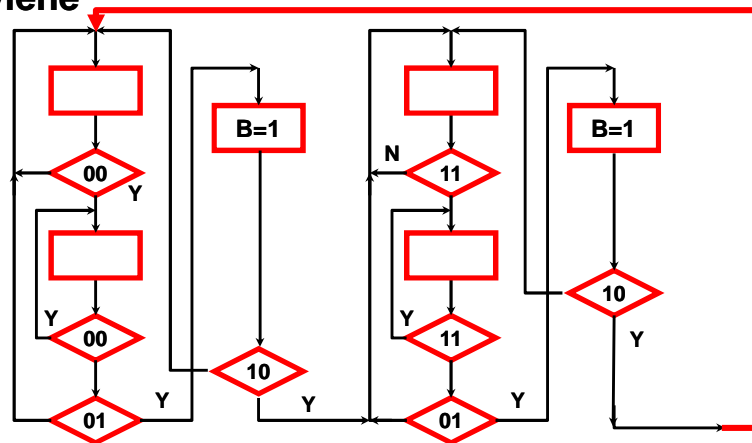


A.S.E.

17.8

Parte di controllo

- Il diagramma ASM delle parte di controllo diviene



A.S.E.

17.9

OSSERVAZIONI

- Sono ora necessari 6 stati interni →
 - 3 variabili di stato + 2 ingressi
- Rete combinatoria sempre troppo grossa
 - Si può provare ad eseguire un precondizionamento degli ingressi

A.S.E.

17.10

Considerazioni sulla sequenza

- Si definiscono due variabili di condizionamento
 - C_0 e C_1
 - La sequenza che inizia con 00 deve essere riconosciuta quando l'uscita è pari $\rightarrow Z_0 = 0$
 - La sequenza che inizia con 11 deve essere riconosciuta quando l'uscita è dispari $\rightarrow Z_0 = 1$
- Si pone
 - $C_0 = 1$ se $X_1, X_0 = 00$ e $Z_0 = 0$ o se $X_1, X_0 = 11$ e $Z_0 = 1$
 - $C_1 = 1$ se $X_1, X_0 = 01$

$\{00-01-10\} _ \{11-01-10\}$

A.S.E.

17.11

Rete Combinatoria di condizionamento

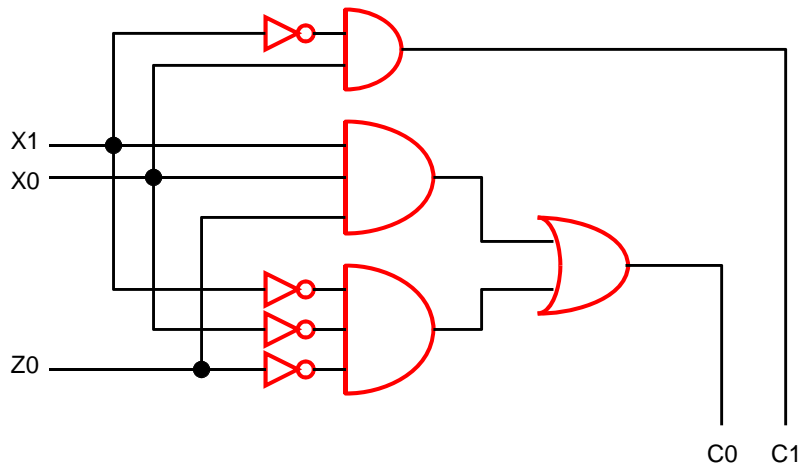
- Tabella di verità

X_1	X_0	Z_0	C_0	C_1
0	0	0	1	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	0

A.S.E.

17.12

Rete Combinatoria di condizionamento



A.S.E.

17.13

Parte di controllo

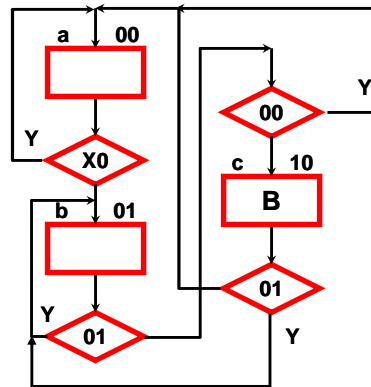
- Ricordando che la parte di condizionamento fornisce
 - $C_0 = 1$ se $X_1, X_0 = 00$ e $Z_0 = 0$ o se $X_1, X_0 = 11$ e $Z_0 = 1$
 - $C_1 = 1$ se $X_1, X_0 = 01$
- Gli ingressi della parte di controllo sono C_0 e C_1
- Le sequenze iniziano quando è $C_0 = 1$
- Il secondo passo della sequenza si ha per $C_1 = 1$
 - Il caso $C_0 \text{ e } C_1 = 1$ non si può mai verificare
- Ora si attiva la parte operativa $B = 1$

A.S.E.

17.14

Diagramma di Flusso della Parte di controllo

- Variabili d'ingresso C_1C_0 , variabili di stato S_1S_0



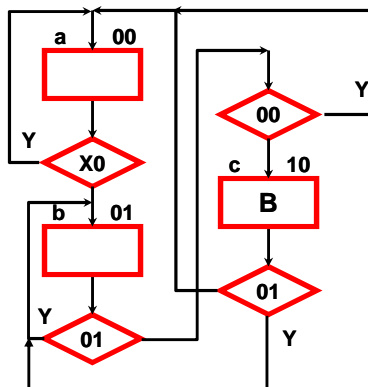
X1	X0	Z0	C1	C0
0	0	0	0	1
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

{00-01-10} _ {11-01-10}

A.S.E.

16.15

Tabella delle transizioni



S1	S0	C1	C0	S1	S0	B
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	0	0
0	0	1	1	--	--	--
0	1	0	0	0	0	0
0	1	0	1	0	1	0
0	1	1	0	1	0	1
0	1	1	1	--	--	--
1	0	0	0	0	0	0
1	0	0	1	0	1	0
1	0	1	0	0	0	0
1	0	1	1	--	--	--
1	1	0	0	--	--	--
1	1	0	1	--	--	--
1	1	1	0	--	--	--
1	1	1	1	--	--	--

A.S.E.

16.16

Minimizzazione

S1	S0	C1	C0	S1	S0	B
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	0	0
0	0	1	1	--	--	--
0	1	0	0	0	0	0
0	1	0	1	0	1	0
0	1	1	0	1	0	1
0	1	1	1	--	--	--
1	0	0	0	0	0	0
1	0	0	1	0	1	0
1	0	1	0	0	0	0
1	0	1	1	--	--	--
1	1	0	0	--	--	--
1	1	0	1	--	--	--
1	1	1	0	--	--	--
1	1	1	1	--	--	--

S _{N1}	S _{N0}	B
0	0	1
0	1	--
1	0	1
1	1	--

S _{N0}	C ₁ C ₀	00	01	11	10
00			1	--	
01			1	--	
11		--	--	--	--
10			1	--	

S _{N1}	C ₁ C ₀	00	01	11	10
00				--	
01				--	1
11		--	--	--	--
10				--	

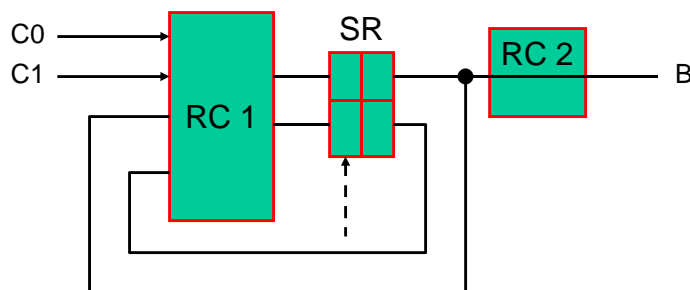
A.S.E.

16.17

Parte di Controllo

• Osservazioni

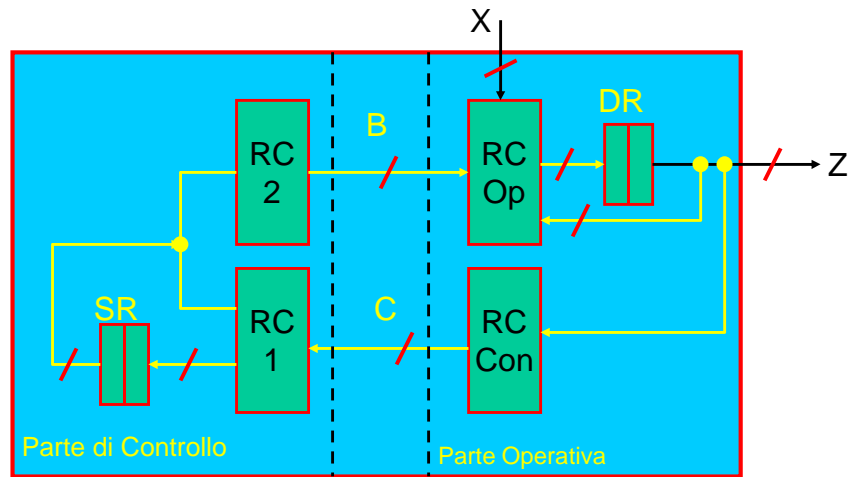
- La variabile B coincide con S_{N1}
- La rete può essere sintetizzata in modo classico come una macchina di MOORE



A.S.E.

17.18

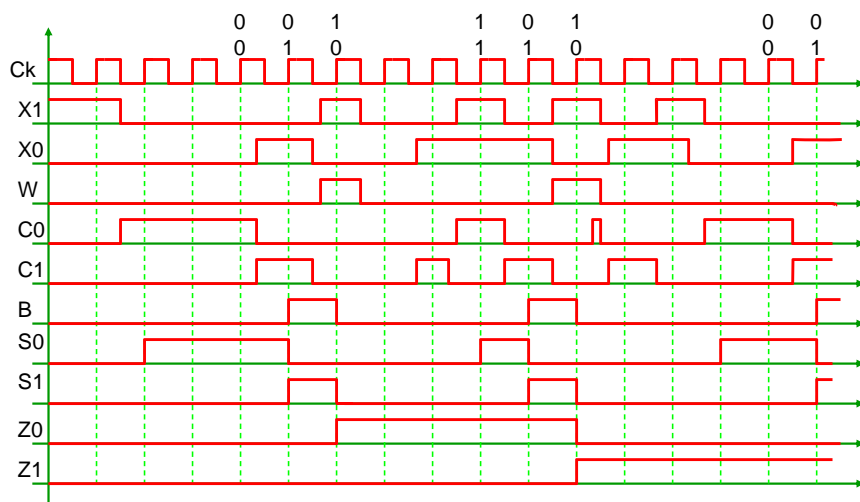
Riconoscitore di Sequenze



A.S.E.

17.19

Temporizzazione



A.S.E.

17.20

Richiami

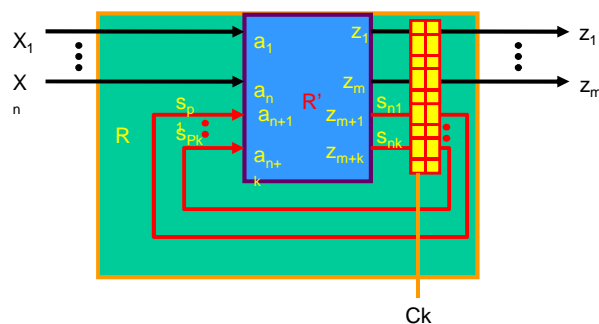
- **Macchina di Mealy ritardata**
- **Registro di stato**
- **Registro operativo**
- **Parte operativa**
- **Parte di controllo**

A.S.E.

17.21

Macchina di Mealy Ritardata

- **Le uscite sono funzioni delle variabili di stato e degli ingressi, ma risultano sincronizzate**
- **Riduce il numero delle variabili di stato**

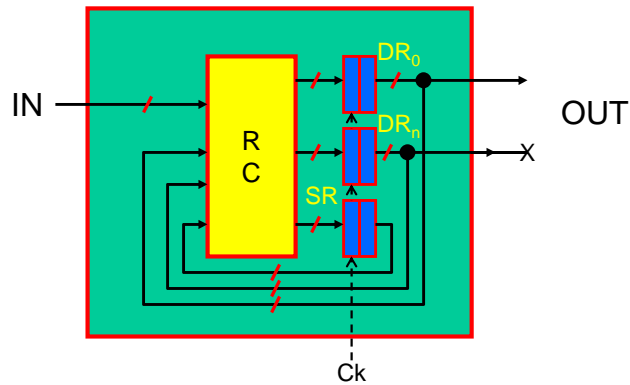


A.S.E.

17.22

Interpretazione diversa

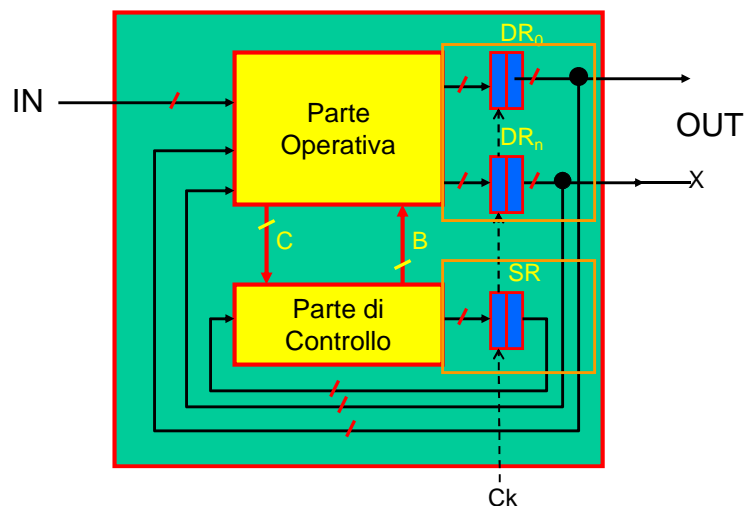
- Il registro può essere visto come più registri che svolgono funzioni diverse



A.S.E.

17.23

Scomposizione della "RC"

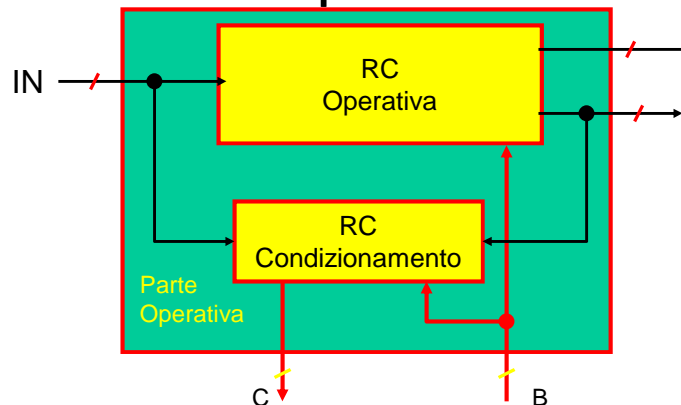


A.S.E.

17.24

Ulteriore suddivisione

- La parte operativa può essere ulteriormente suddivisa in due parti



A.S.E.

17.25

Osservazioni

- La suddivisione vista è dettata da:
 - Le reti sequenziali complesse sono difficili ad essere gestite
 - Le funzioni della parte operativa solitamente sono.
 - OPERAZIONI ARITMETICHE
 - OPERAZIONI LOGICHE
 - SHIFT
 - MULTIPLEX
 - Una soluzione “guidata” è solitamente più veloce e più efficiente
 - (i criteri d’ottimizzazione sono scelti dal progettista)

A.S.E.

17.26

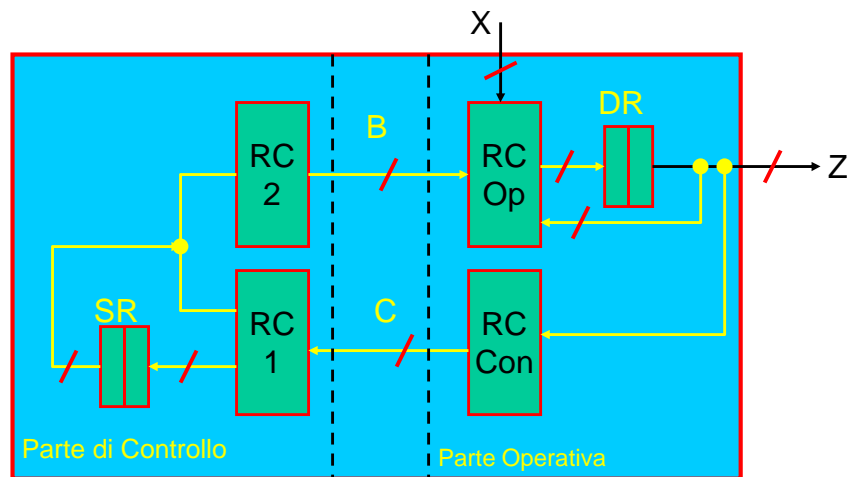
Esempio

- **Riconoscitore di due sequenze in cascata**
 - Ingressi
 - X_1, X_0
 - Uscite
 - Z_3, Z_2, Z_1, Z_0
- **Descrizione**
 - prima riconosce la sequenza 00, 01, 10
 - dopo riconosce la sequenza 11, 01, 10
 - Incrementa l'uscita tutte le volte che una sequenza viene riconosciuta (modulo 16)

A.S.E.

17.27

Riconoscitore di Sequenze



A.S.E.

17.28

Requisiti della Parte Operativa

- Funzioni che deve eseguire fra 2 parole di k bit
 - Somma
 - Differenza
 - Negazione
 - And
 - Or
 - Shift
 -
- Possibilità di essere “PROGRAMMATA”
- Eventuale memorizzazione del risultato

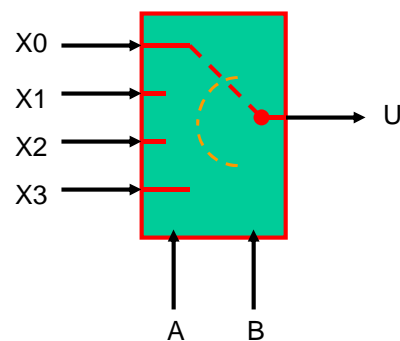
A.S.E.

17.29

Multiplex 4 a 1

- 4 ingressi X0, X1, X2, X3
- 2 segnali di controllo A, B

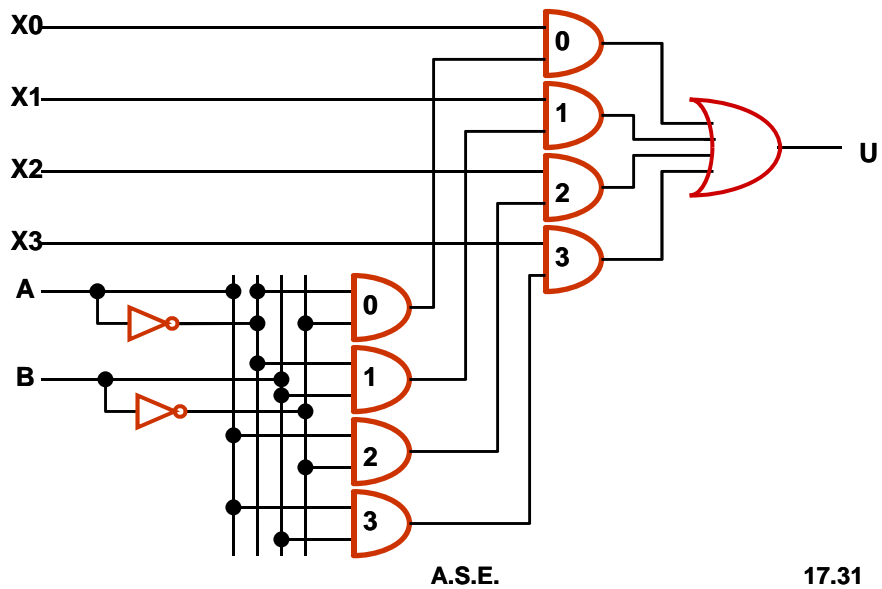
A	B	U	— —
0	0	X0	$\overline{A} \cdot \overline{B}$
0	1	X1	$\overline{A} \cdot B$
1	0	X2	$A \cdot \overline{B}$
1	1	X3	$A \cdot B$



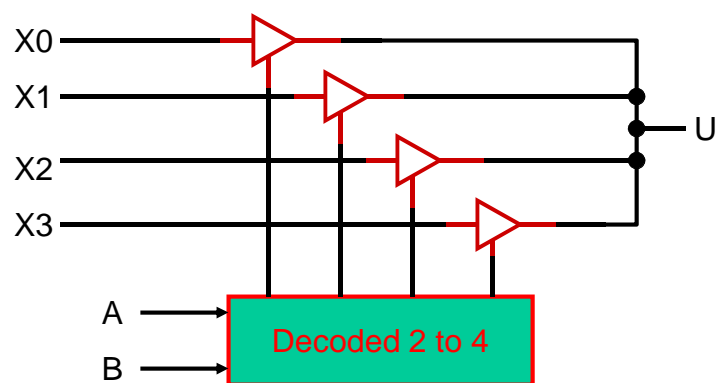
A.S.E.

17.30

Soluzione 1

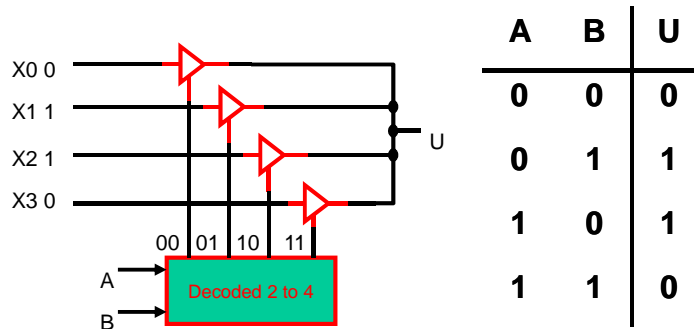


Soluzione 2



Interpretazione diversa 1

- Ingressi A e B controllo X_0, X_1, X_2, X_3



Programmazione $X = 0110$

$$U = A \oplus B$$

A.S.E.

17.33

Interpretazione diversa 2

- A e B ingressi, $X_0 - X_3$ variabili di controllo
- Primi 8 casi

N	X3	X2	X1	X0	U	U'	A	B	U	---
0	0	0	0	0	0	0	0	0	X0	$\overline{A \cdot B}$
1	0	0	0	1	$\overline{A \cdot B}$	$\overline{A \cdot B}$	0	1	X1	$\overline{A \cdot B}$
2	0	0	1	0	$\overline{A \cdot B}$	$\overline{A \cdot B}$	1	0	X2	$\overline{A \cdot B}$
3	0	0	1	1	$\overline{A \cdot B} + \overline{A \cdot B}$	$\overline{A \cdot B}$	1	1	X3	$A \cdot B$
4	0	1	0	0	$\overline{A \cdot B}$	$\overline{A \cdot B}$				
5	0	1	0	1	$A \cdot \overline{B} + \overline{A \cdot B}$	\overline{B}				
6	0	1	1	0	$\overline{A \cdot B} + \overline{A \cdot B}$	$A \oplus B$				
7	0	1	1	1	$\overline{A \cdot B} + \overline{A \cdot B} + A \cdot B$	$\overline{A \cdot B}$				

A.S.E.

17.34

Interpretazione diversa 3

- A e B ingressi, X0 – X3 variabili di controllo
 - Secondi 8 casi

N	X3	X2	X1	X0	U	U'
8	1	0	0	0	$A \cdot B$	$\overline{A \cdot B}$
9	1	0	0	1	$A \cdot B + \overline{A} \cdot \overline{B}$	$A \oplus B$
A	1	0	1	0	$A \cdot B + \overline{A} \cdot B$	\overline{B}
B	1	0	1	1	$A \cdot B + \overline{A} \cdot B + \overline{A} \cdot \overline{B}$	$\overline{A} + B$
C	1	1	0	0	$A \cdot B + A \cdot \overline{B}$	A
D	1	1	0	1	$A \cdot B + A \cdot \overline{B} + \overline{A} \cdot \overline{B}$	$A + \overline{B}$
E	1	1	1	0	$A \cdot B + A \cdot \overline{B} + \overline{A} \cdot B$	$A + B$
F	1	1	1	1	1	1

A.S.E.

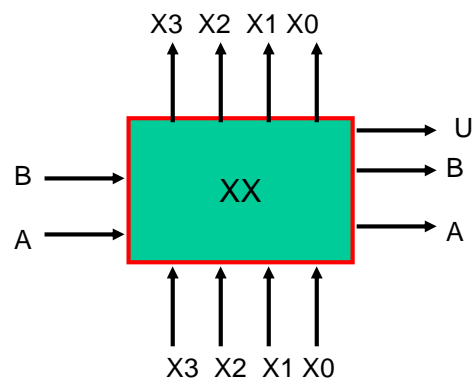
17.35

N	X3	X2	X1	X0	U	U'
0	0	0	0	0	0	0
1	0	0	0	1	$\overline{A} \cdot \overline{B}$	$\overline{A \cdot B}$
2	0	0	1	0	$\overline{A} \cdot B$	$\overline{A \cdot B}$
3	0	0	1	1	$\overline{A} \cdot B + \overline{A} \cdot B$	\overline{A}
4	0	1	0	0	$A \cdot \overline{B}$	$A \cdot \overline{B}$
5	0	1	0	1	$A \cdot \overline{B} + \overline{A} \cdot \overline{B}$	\overline{B}
6	0	1	1	0	$\overline{A} \cdot B + \overline{A} \cdot B$	$A \oplus B$
7	0	1	1	1	$\overline{A} \cdot B + \overline{A} \cdot B + \overline{A} \cdot \overline{B}$	$\overline{A \cdot B}$
8	1	0	0	0	$A \cdot B$	$\overline{A \cdot B}$
9	1	0	0	1	$A \cdot B + \overline{A} \cdot \overline{B}$	$A \oplus B$
A	1	0	1	0	$A \cdot B + \overline{A} \cdot B$	\overline{B}
B	1	0	1	1	$A \cdot B + \overline{A} \cdot B + \overline{A} \cdot \overline{B}$	$\overline{A} + B$
C	1	1	0	0	$A \cdot B + A \cdot \overline{B}$	A
D	1	1	0	1	$A \cdot B + A \cdot \overline{B} + \overline{A} \cdot \overline{B}$	$A + \overline{B}$
E	1	1	1	0	$A \cdot B + A \cdot \overline{B} + \overline{A} \cdot B$	$A + B$
F	1	1	1	1	1	1

A.S.E.

17.36

Blocco programmabile



A.S.E.

17.37

Carry Look - Ahead Adder

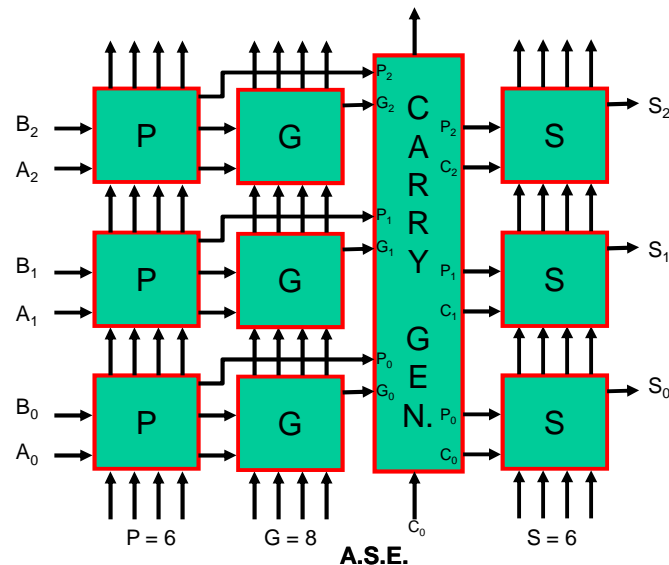
- Definite le funzioni

- **Generate** $G_i = A_i B_i$
- **Propagate** $P_i = A_i \oplus B_i$
- Risulta
- **Somma** $S_i = P_i \oplus C_i$
- **Carry** $C_{i+1} = G_i + P_i \cdot C_i$

A.S.E.

17.38

Sommatore a 3 bit



17.39

Osservazioni 1

- Struttura modulare PROGRAMMABILE
- Bit di programmazione:
- C_0 P G S
- Alcuni esempi

Co	P	G	S	Funzione
0	6	8	6	A+B
1	9	4	6	A-B
1	9	2	6	B-A
0	8	0	A	A and B
0	E	0	A	A or B

A.S.E.

17.40

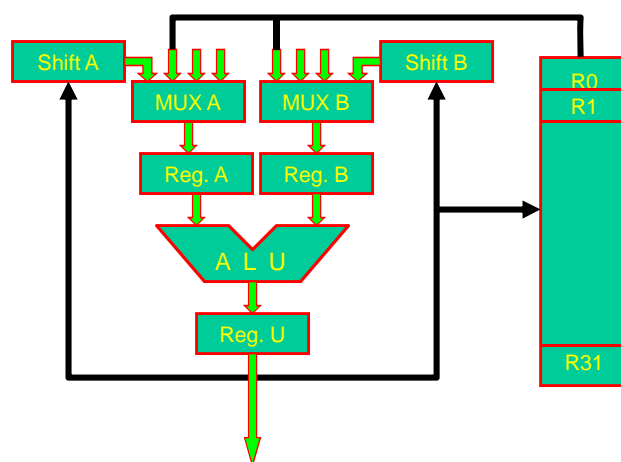
Osservazioni 2

- L'unità realizzata esegue sia funzioni logiche che aritmetiche
- ALU (Arithmetic Logic Unit) Programmabile
- Bit di programmazione 13
- Non tutte le combinazioni hanno significato
- Si possono aggiungere dei MUX per instradare sia gli ingressi che l'uscita
- Si possono aggiungere dei registri per memorizzare i risultati
- Si può aggiungere uno shift register per moltiplicare o dividere per 2

A.S.E.

17.41

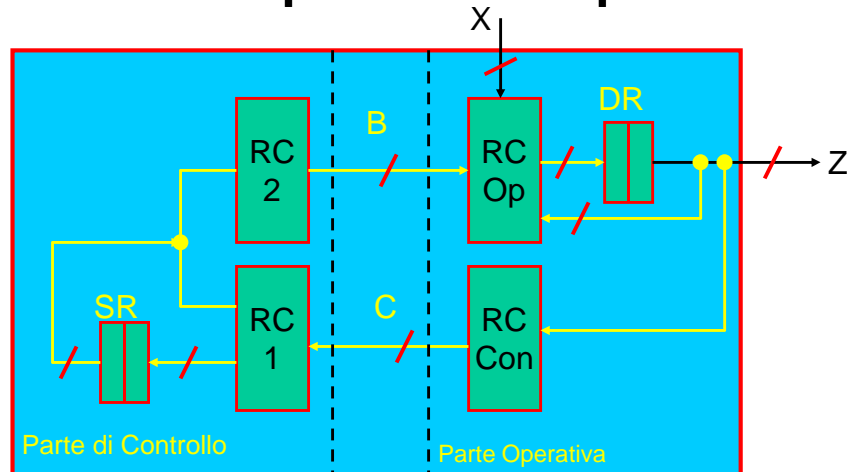
Parte Operativa



A.S.E.

17.42

Architettura di Rete sequenziale complessa



A.S.E.

17.43

Osservazioni

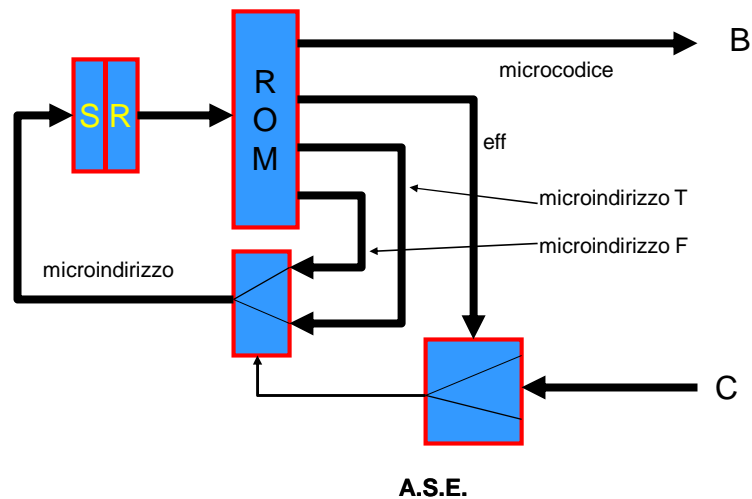
- **La parte operativa ha una soluzione generale**
 - Rete combinatoria operativa (ALU)
 - Rete combinatoria di condizionamento (Carry etc.)
- **Soluzione non ottimizzata per la particolare esigenza**
- **La parte di controllo è ottimizzata**
- **Si può trovare soluzioni più generali non ottimizzate**
- **Una rete combinatoria si può sempre realizzare con una ROM**

A.S.E.

17.44

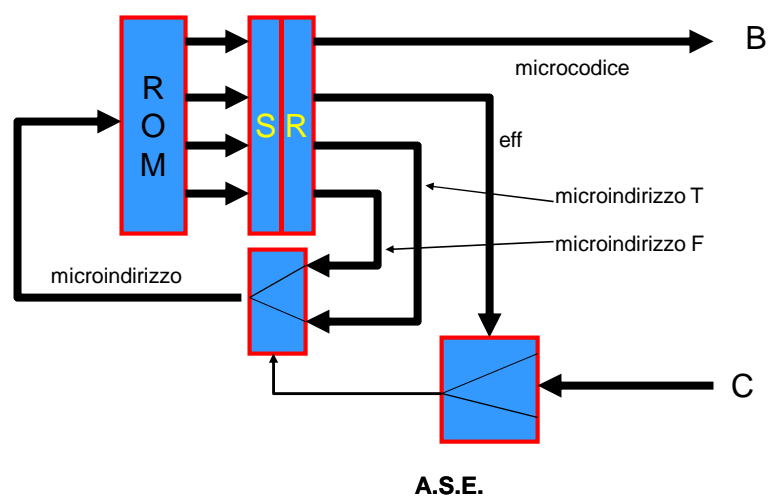
Parte di controllo 1

- *Microaddress-based*



Parte di controllo 2

- *Microinstruction-based*



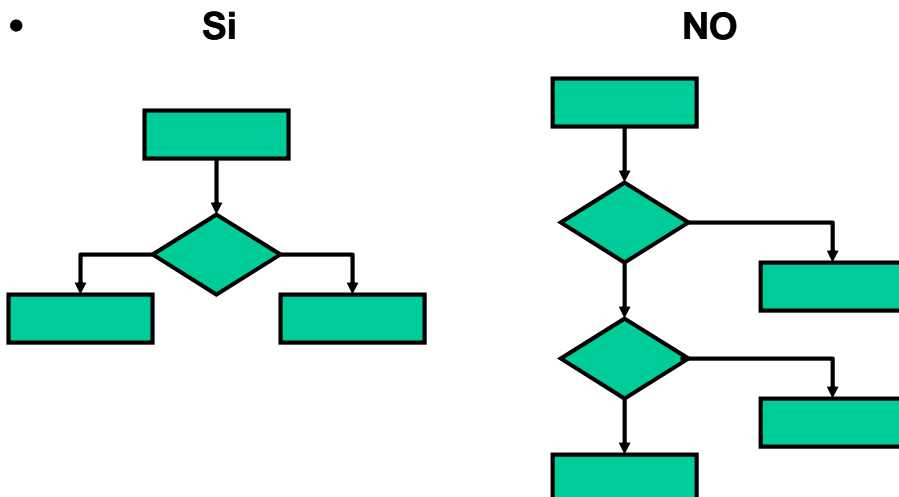
Osservazioni

- **Microaddress-based**
 - il registro di stato è piccolo
 - la ROM è in serie alla parte operativa
- **Microinstruction-based**
 - il registro di stato è grande
 - fra ROM e parte operativa c'è il registro di stato
- **Consente la presenza di un solo blocco di decisione fra due stati contigui**

A.S.E.

17.47

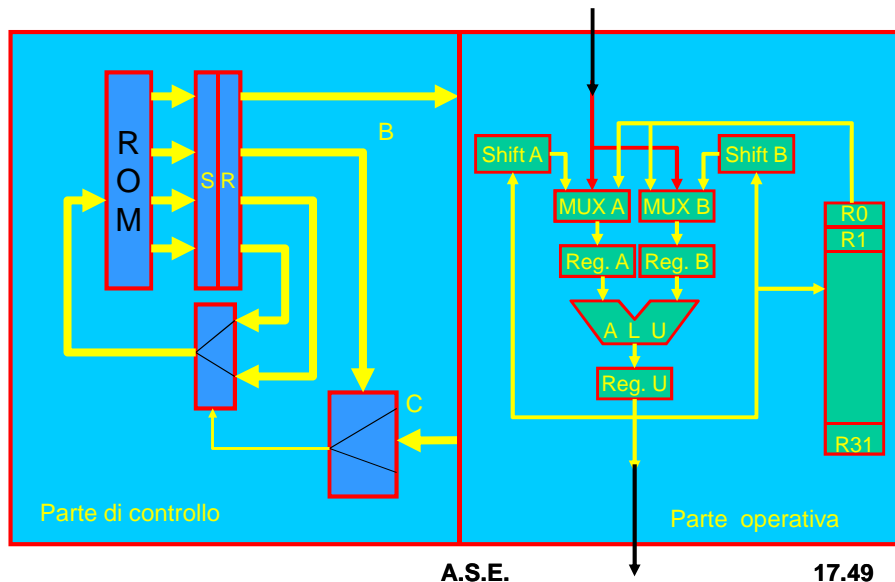
Osservazione sui microsalti



A.S.E.

17.48

Architettura Tot.



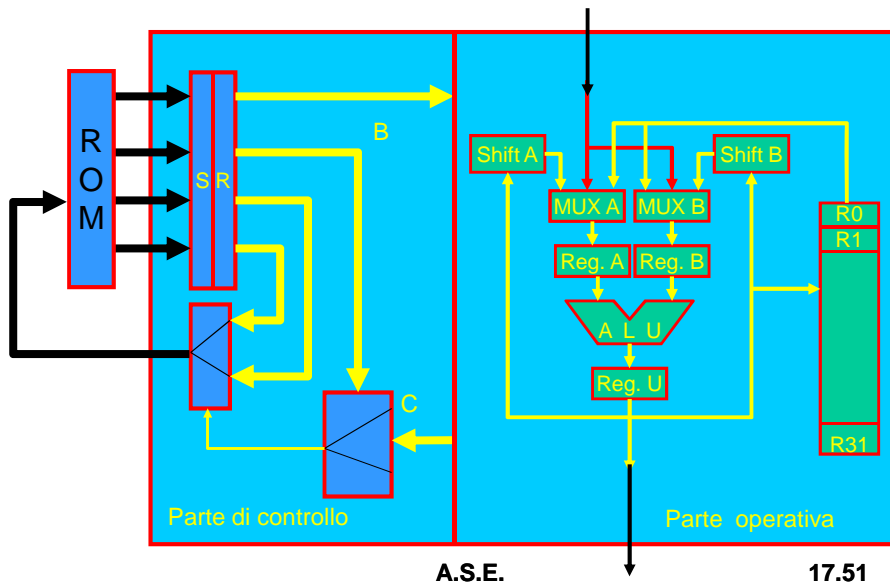
Osservazioni

- L'architettura della parte di controllo *Microinstruction-based* evita di avere due reti combinatorie in cascata
- La parte operativa è *general purpose*
- La parte di controllo ha una architettura generale, ma è progettata per la particolare applicazione
- L'elemento che personalizza l'applicazione è la ROM

A.S.E.

17.50

Architettura modificata



Osservazioni 1

- La nuova architettura consente di personalizzare il sistema per varie applicazioni
- **Inconvenienti**
 - La memoria di personalizzazione, la ROM, presenta una lunghezza di parola eccessiva
 - Microcodice (> di 100 bit)
 - Microindirizzi T (> di 10 bit)
 - Microindirizzi F (> di 10 bit)
 - Condizionamento *eff* (8 bit)

A.S.E.

17.52

Osservazione 2

- **Tecniche di realizzazione di una rete logica**
 - **Logica cablata**
 - Viene progettata una particolare architettura (ottimizzata) per la risoluzione dello specifico problema
 - Si usano blocchi base standard, le interconnessioni (cablaggio) realizzano la particolare architettura
 - **Logica a controllo di programma**
 - Viene progettata una architettura che è in grado di svolgere varie funzioni
 - La personalizzazione è ottenuta mediante la particolare sequenza di operazioni svolte dalla macchina
 - Programma

A.S.E.

17.53

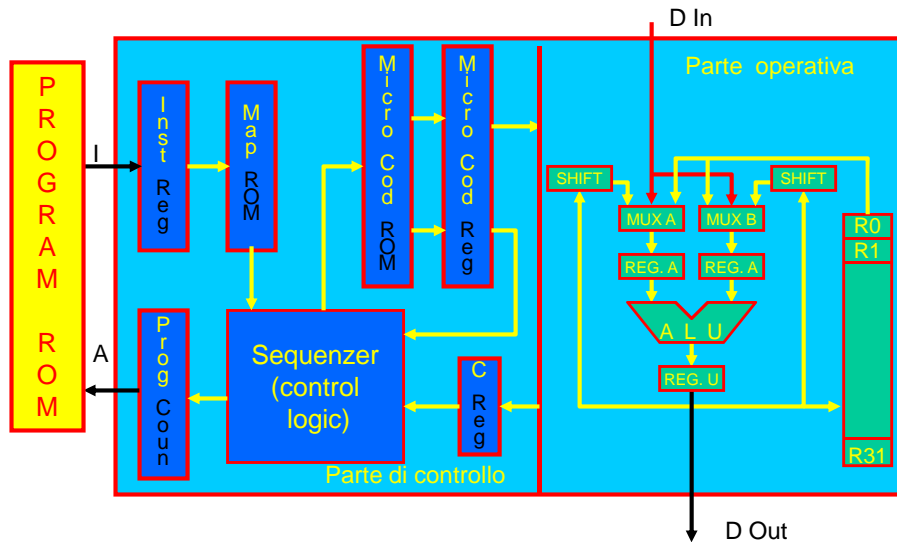
Osservazione 3

- Si mappano nella ROM di microprogramma tutte le funzioni possibili della parte operativa
- Si codificano in modo compatto le varie funzioni in una ROM di mappatura
- Il programma di personalizzazione dell'applicazione, scritto in un linguaggio opportuno (ASSEMBLER), risiede in una memoria esterna
- Sono necessari alcuni altri registri e una opportuna rete di controllo (SEQUENZIALIZZATORE)

A.S.E.

17.54

Architettura a controllo di programma



A.S.E.

17.55

Osservazioni

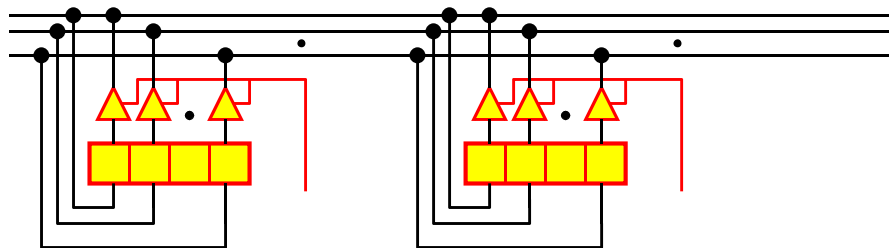
- **Sequenzializzatore**
 - Gestisce sia il microprogramma, sia il programma
- **Program Counter**
 - Contiene l'indirizzo *successivo* della locazione di memoria di programma
- **Status Register**
 - Contiene informazioni sulle funzioni eseguite dalla parte operativa
- **Inconveniente**
 - *Elevato numero di terminali (PED)*

A.S.E.

17.56

BUS

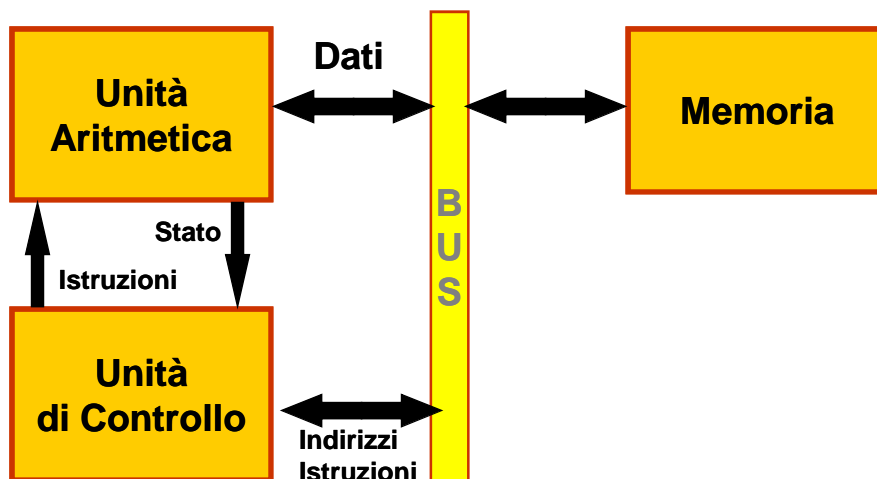
- Utilizzando porte TRE STATE bidirezionali si può utilizzare lo stesso *Filo* per collegare, in istanti diversi, vari componenti
- **BUS** Percorso informatico condiviso fra più unità



A.S.E.

17.57

Architettura von Neumann



A.S.E.

17.58

Descrizione

- **Unità Aritmetica:**
 - **ALU**
 - Blocco che “esegue” le operazioni richieste
- **Unità di Controllo:**
 - **MSF (+ eventuali registri)**
- **Memoria**
 - **Unica per Dati e Programma (RAM)**

A.S.E.

17.59

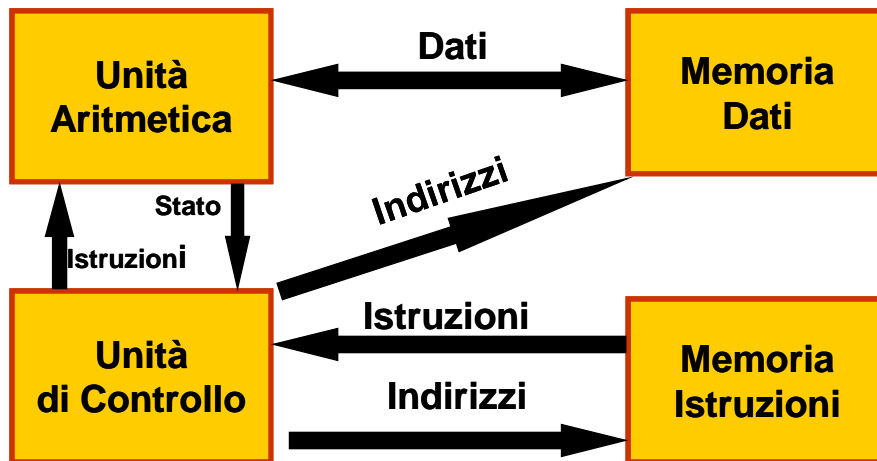
Osservazioni

- **Macchina di ridotta complessità**
=====
- **Criterio di valutazione di una architettura**
 - Individuazione dei blocchi “congestionati”
=====
- **Limiti**
 - Unica memoria
 - Unico BUS (Collo di Bottiglia)

A.S.E.

17.60

Architettura Harvard



A.S.E.

17.61

Descrizione

- **Memoria Dati**
 - » Accesso diretto alla ALU
 - » Realizzazione: RAM
- **Memoria Istruzioni**
 - » Il contenuto rimane invariato
 - » Realizzazione con memoria non volatile: ROM

A.S.E.

17.62

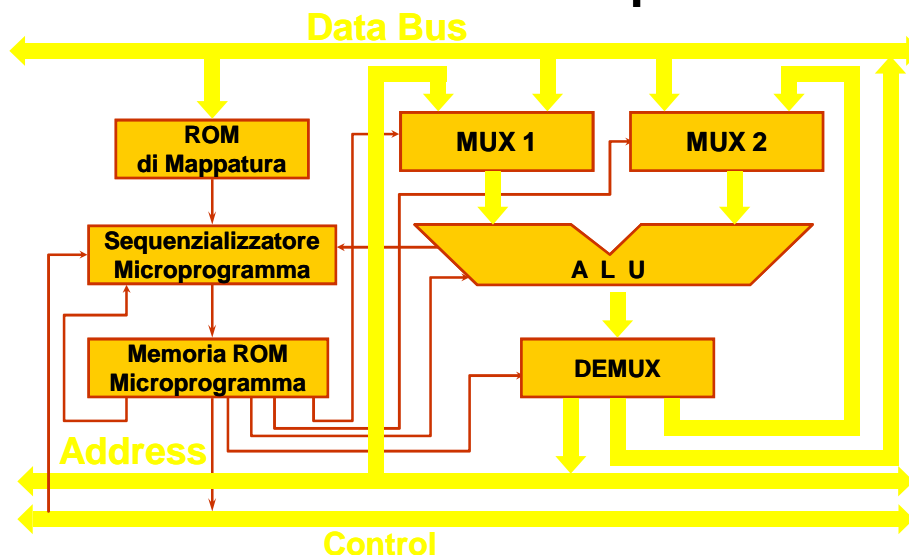
Osservazioni

- Non presenta particolari “Colli di bottiglia”
- Può essere migliorata per aumentarne la concorrentialità
- Architettura alla quale si ispirano gli attuali microprocessori e DSP

A.S.E.

17.63

Architettura di un Microprocessore



A.S.E.

17.64

Descrizione (1)

- **ROM di Mappatura**
 - Converte il codice macchina in codice interno
- **Sequenzializzatore**
 - Rete di controllo effettiva
- **Memoria di Microprogramma**
 - Converte il codice interno nei codici di controllo dei vari blocchi

A.S.E.

17.65

Descrizione (2)

- **ALU**
 - Unità Logica Aritmetica
- **MUX 1 e MUX 2**
 - Multiplex di scelta su gli ingressi della ALU
- **DEMUX**
 - Gestisce l'indirizzamento del risultato

A.S.E.

17.66

Osservazioni

- **Funzionamento totalmente in sequenza**
 - Interpretazione → Esecuzione
- **Assenza di registri interni**
 - Contatore di programma, Registro di stato
 - Accumulatori
- **Generazione degli indirizzi mediante ALU**
- **Collo di bottiglia → DATA BUS**

A.S.E.

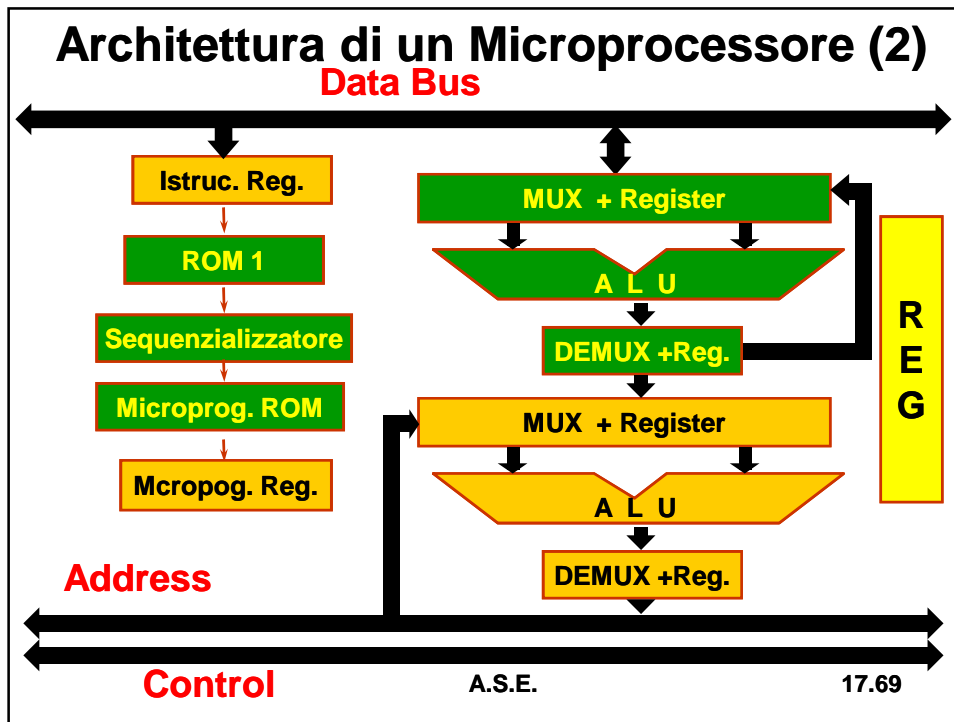
17.67

PIPE-LINE

- **Alcune operazioni possono essere fatte in parallelo**
 - Necessità di disporre di REGISTRI intermedi
- **Mentre la ALU esegue le operazioni richieste, il Sistema di controllo può decodificare l'istruzione successiva**
- **Aggiungere una ALU per il calcolo degli indirizzi**

A.S.E.

17.68



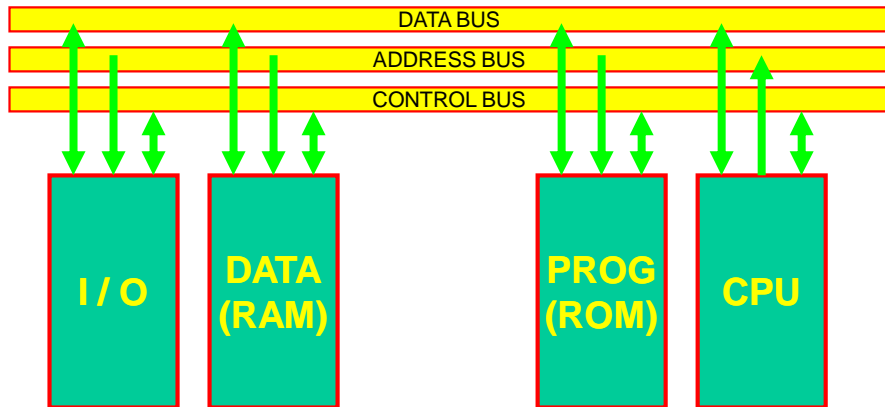
Osservazioni

- Possibilità di sovrapposizione fra fase di interpretazione ed esecuzione dell'istruzione
- Possibilità di memorizzazione di risultati intermedi
- Salvataggio dei registri di controllo
 - Possibili ulteriori modifiche in funzione delle esigenze

A.S.E.

17.70

Architettura di un calcolatore



A.S.E.

17.71

Tipi di architetture in funzione delle istruzioni

- **RISC => Reduced Instructions Set Computer**
 - ridotto numero di istruzioni
 - si effettuano in un numero ridotto di cicli
- **CISC => Complex Instructions Set Computer**
 - Istruzioni di elevata capacità
 - si effettuano in un numero grande di cicli

A.S.E.

17.72

Microcontrollori

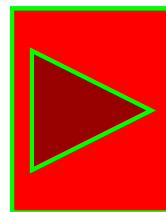
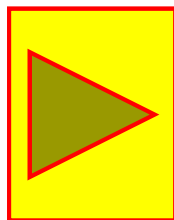
- Un intero computer su singolo Chip
- Elementi presenti
 - CPU
 - RAM dati
 - PROM di programma (Flash)
 - PROM dati (E²PROM)
 - Terminali di I/O indipendenti
 - Temporizzatori
 - PWM
 - Watchdog Timer

A.S.E.

17.73

Esempio

- ATMEL
- ATmega8515 ATmega169



A.S.E.

17.74

Conclusioni

- Reti sequenziali sincronizzate complesse
- ALU
- Parte di controllo
 - *Microaddress-based*
 - *Microinstruction-based*
- Architetture della parte di controllo
- Architetture di un processore
- Bus
- Architettura Von Neuman
- Architettura Harvard
- Architettura di un CALCOLATORE
- Microcontrollore

A.S.E.

17.75

Conclusioni

- Reti sequenziali sincronizzate complesse
 - Macchina di Mealy ritardata
 - Registro di stato
 - Registro operativo
 - Parte operativa
 - Parte di controllo
- Esempio

A.S.E.

17.76